Claim Rejections Under 35 U.S.C. §103

I. Claims 1-34 Stand Rejected Under 35 U.S.C. §103

Claims 1-34 stand rejected under 35 U.S.C. §103. Independent claims 1, 12 and 23 stand rejected as being unpatentable over Cantin et al (EP 0 690 375 A2) ("Cantin") in view of Japanese patent application assigned to NEC Corp., Japanese Patent Application No. 1997JP-0303475 ("NEC"). Applicants respectfully traverse this rejection and contend that claims 1-34 are patentable and in condition for allowance.

For ease of discussions, summaries of the claimed invention, the primary reference of Cantin, and secondary reference of NEC are provided below.

A. Summary of Claimed Invention

The claimed invention is directed towards determining what method of an object to call in an object-oriented environment from a technical computing environment by calculating a ranking of method signatures corresponding to the method. A method signature is not an object or a method or a programming instruction, but a character expression representing the specification of the interface of the method. The characters of the method signature's expression describe what data elements the method expects or needs to know about. Each method signature may comprise the method's name and the number and type of each input and output parameter of the method. For example, the java method of "void sampleMethod (int arg1, double arg2, java.lang.string arg3) may have a method signature of "(IDLjava/lang/string;)V". As shown by the example, Java, as with other programming languages, have a vocabulary for specifying method signatures. For example, in the above example and in accordance with Java's vocabulary for method signatures, the letter I in "IDL" of the java method signature for

sampleMethod represents an integer data type, and the letter D a double data type. The method signature itself is not a method that can be executed or invoked, and does not contain any program instruction or computer memory pointer to invoke the method to which it corresponds. It is used to uniquely identify a method and represent the specification of the method's interface by the vocabulary of the method signature language.

Within a class in an object-oriented environment, each method having the same name must have a different number of inputs, or one or more inputs must differ by data type. Since a method signature represents the number and types of inputs to a method, each method having the same name within that class will have a unique method signature. As such, method signatures can be used to uniquely identify the method that should be called when there are multiple methods with the same name that could be called.

In a technical computing environment, most of the data types are represented as arrays of multiple dimensions. Array-based data types do not distinguish between a scalar, vector or a matrix data type. Because the technical computing environment uses array-based data, it is difficult to invoke methods of objects in an object oriented environment that have the same name and are only distinguished by the data types of their input parameters. For example, the technical computing environment may have its own data type that is not available in the object-oriented environment. As such, an input parameter received from a technical computing environment for a method call on an object may not fit agreeably into a data type of the object-oriented environment.

In order to determine an appropriate method to call, the claimed invention compares the technical computing environment data to be provided as input to the method with the specified data types expected as described in the method signature. The method signatures are ranked

based on which corresponding methods are better suited to accept the input parameters of the data from the calling array-based computing environment. Based on the comparison, the claimed invention automatically selects a method signature according to the ranking and then invokes the method corresponding to the selected method signature.

B. Summary of Cantin

The method of Cantin is directed towards having a standard way to implement the mapping of object data to a permanent storage medium in view of the Object Persistence Service Specification (OPSS) as set by the Object Management Group, Inc. Cantin describes the mapping of data contained in an active instance of an object to a permanent storage medium, such as a database. Cantin discusses a specialized implementation of a persistence data service ("PDS") to receive an instance of an object and then store the values set in the properties of the received object to a database table associated with the object. An abstract class is defined in the persistence storage service having custom methods that implement the specific program instructions to store and retrieve the object data to and from an associated database table.

Specifically, Cantin creates a specialized version of an AbstractSchemaMapper base class for each object that is to be stored in a persistent medium. The AbstractSchemaMapper class has four methods to be customized for a given class: SchemaMapperStoreNew, SchemaMapperStoreExisting, SchemaMapperRestore and SchemaMapperDelete. For each of these four methods, custom code is written to map the specific object properties of the object to the specific column names in the database table where the properties will be stored. These methods do not handle persistent storage of any methods of the object.

For example, the method SchemaMapperStoreNew of the AbstractSchemaMapper can be customized to store a new account object to a database. The account object may have the properties of Account_id, Branch_Id and Account_Balance. For an active instance of the account object, these properties will be set to values for that specific instance. The SchemaMapper methods above would be customized to store, update, restore and delete the property values of the account object to a database. For example, there may be an account table in the database with the columns of AccountId, BranchId and AccountBalance. The SchemaMapperStoreNew method will take as input an instance of a new account object, take the values for Account_id, Branch_Id and Account_Balance from the object's properties, and then execute a sql statement to insert a row of data with these values in the account table in the database. As such, Cantin maps an object to a persistent medium, like a database, by customizing the four SchemaMapper methods to handle the specifically named properties of the object.

C. Summary of NEC

NEC is directed toward generating a database index key comprising a composite index key to improve search methods for objects in an object-oriented database management system. A database index is used to locate data in a database. A database index allows a search of the database for a specific instance of data, e.g. a row of data having a specific key value, without searching the entire database, much like using an index for an instruction manual. A search looks at the index to find the location of the data and then obtains the data from that location. Database indexing is used to improve database searching performance.

NEC describes a method for creating a composite index key from a data structure stored in the database representing property values of an object. The composite key index then becomes the basis of a management object subsequently used to retrieve objects. For example, a data structure for employee information may contain two variables representing the name of the employee and affiliation coding. These variables represent the properties of an object stored in the tables of the database. For indexing purposes, the variables of the data structure comprising the composite index key are ranked by size. The size would indicate the size of the data the variables can hold.

D. Independent Claims 1 and 12 Stand Rejected Under 35 U.S.C. §103

Independent claims 1 and 12 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC. Claims 1 and 12 are independent claims. Applicants respectfully traverse this rejection.

Claims 1 and 12 are directed to a method and computer-readable medium claim, respectively. These claims recite a method for *retrieving a set of method signatures* and comparing the data types of the *signatures* to data types of input parameters received from an array-based computing environment. These claims further recite ranking the *method signatures* as a function of the comparison, selecting one of the *method signatures* according to the ranking, and invoking the method corresponding to the selected *method signature*. Cantin in view of NEC does not teach or suggest each and every feature recited in these claims.

Cantin does not teach or suggest *retrieving a set of method signatures* as recited in independent claims 1 and 12. In the office Action, the Examiner argues that Cantin teaches *retrieving a set of method signatures* because Cantin discusses retrieving an object having

properties and methods. Applicants respectfully contend that retrieving an object is not comparable to *retrieving a set of method signatures*. As discussed in the Summary of the Claimed Invention above, a method signature is a character expression of the name and data types of a method to uniquely identify a method. A method signature is not an object, method or programming instruction. Cantin does not discuss *retrieving method signatures*. Rather, Cantin discusses retrieving the values of properties of an instance of an object from a database table. Cantin does not retrieve *method signatures*, or even methods, from the database table. As such, Cantin fails to teach or suggest *retrieving a set of method signatures.*

Furthermore, Cantin does not teach or suggest comparing the data types of the *signature* to data types of input parameters received from an array-based computing environment as recited in independent claims 1 and 12. Since Cantin does not retrieve *method signatures*, Cantin therefore cannot be comparing data types of a *signature*. However, in the office Action, the Examiner equates the mapping between a selected object and a persistent data of Cantin to comparing the data types of the *signature* to the data types of input parameters as in the claimed invention. Applicants respectfully disagree with the Examiner and contend that the mapping the Examiner cites is not comparable to comparing the data types of the *signature*. As discussed above, a method signature is a character expression representing the method name and data types of the method. The claimed invention compares the data types of the *method signature* to the data types of input parameters received from an array-based computing environment. Cantin does not discuss comparing the data types of the arguments to a *method signature* to the data types received as input for those arguments. Cantin is not concerned with *method signatures*, or with comparing the data types of arguments of a *method signature.* Rather, Cantin calls specialized methods to store the object to a database table. For each value of a property of the

object to be stored to the database, the specialized method writes the property values to the database table. As such, Cantin is focused on storing properties of an object to a database table. Therefore, Cantin fails to teach or suggest comparing data types of a *method signature* to data types of input parameters received from an array-based computing environment.

Examiner admits in the office Action that Cantin does not teach or suggest the recited claim limitations of ranking the *method signatures* as a function of the comparison, and selecting one of the *method signatures* according to the ranking. The Examiner cites NEC for the purpose of suggesting one ordinarily skilled in the art might modify Cantin to rank *method signatures* as a function of the comparison, and selecting one of the *method signatures* according to the ranking.

NEC does not discuss retrieving a set of *method signatures*, or comparing the data types of the *signature* to data types of input parameters received from an array-based computing environment as recited in independent claims 1 and 12. NEC discusses generating a database index key from variables in a database. As discussed above, Cantin fails to teach or suggest retrieving a set of *method signatures*, or comparing the data types of the *signature* to data types of input parameters received from an array-based computing environment. Therefore, NEC fails to bridge the factual deficiencies of Cantin with regards to these claim limitations.

Furthermore, Applicants content that NEC fails to teach or suggest the claim limitations of ranking the *method signatures* as a function of the comparison and selecting one of the *method signatures* according to the ranking. The comparison recited in this limitation is the comparison of the data types of the *signature* to the data types of received input parameters from an array-based computing environment. NEC discusses ranking the size of variables of data structures stored in the database for using as a composite index key for searching the database. Ranking

the size of a variable of a data structure is not comparable to ranking the data type of a method signature in comparison to the data type of an input parameter. As such, NEC compares sizes of variables, and does not compare data types as in the claimed invention.

For at least the aforementioned reasons, Applicants submit that Cantin in view of NEC does not detract from the patentability of independent claims 1 and 12. Claims 2-11 depend on and incorporate the patentable subject matter of independent claim 1, and claims 13-22 depend on and incorporate the patentable subject matter of independent claim 12. As such, Applicants submit that Cantin in view of NEC does not detract from the patentability of dependent claims 1 and 12. Accordingly, Applicants respectfully request the withdrawal of the Examiner's rejection of claims 1-22 under 35 U.S.C. §103.

E. Independent Claim 23 Stands Rejected Under 35 U.S.C. §103

Independent claim 23 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC. Applicants respectfully traverse this rejection.

Claim 23 recites a system comprising an object-oriented environment and a technical computing environment. The object-oriented environment includes an interface for identifying methods provided by objects. The *technical computing environment* comprises a *calculation workspace*, a command interpreter, and a *signature selector*. The signature selector accesses the interface of the object-oriented environment to retrieve and rank a list of *signatures* corresponding to methods defined within the object-oriented environment. Cantin in view of NEC does not teach or suggest each and every feature recited in claim 23.

Cantin does not teach or suggest the *technical computing environment* of the claimed environment. In the office Action, the Examiner equates the persistent data service ("PDS") of

Cantin with the *technical computing environment* of the claimed invention. Applicants contend that the PDS does not provide a technical computing environment. That is, the PDS of Cantin is a service that provides a set of methods to store and retrieve object properties to a database (see Cantin, page 2, lines 26-30). Cantin does not discuss the PDS as providing an environment, such as one provided by a mathematic tool, by which technical computing activities can be performed. Furthermore, Cantin does not teach or suggest the *technical computing environment* comprising a *calculation workspace* and a *signature selector*. In the office Action, the Examiner equates the SchemaMapper of Cantin with the *calculation workspace* of the claimed invention. The SchemaMapper does not provide a workspace for technical computing calculations in a technical computing environment. The SchemaMapper maps the properties of an object to its column name in a relational database (see Canting, page 2, lines 40-44). In the office Action, the Examiner equates the schema mapping methods of the PDS as a *signature selector*. The schema mapping methods and the PDS do not select *method signatures*. As previously discussed, a *method signature* is a character expression of the method name and data types of the method, and is separate from the method itself or any programming instructions of the method. Cantin is focused on storing object properties to a database and does not teach or suggest any structure or methods for retrieving a list of *method signatures*. Hence, the apparatus of Cantin teaches and suggests a structure, and an operation, and a function different from the claimed invention.

The Examiner cites NEC for the purpose of suggesting one ordinarily skilled in the art might modify Cantin to provide a signature selector to rank a list of signatures corresponding to methods. NEC discusses ranking the size of variables of data structures stored in the database for using as a composite index key for searching the database. NEC does not discuss a *signature*

selector retrieving and ranking a list of *signatures* corresponding to methods. As such, NEC fails to bridge the factual deficiencies of Cantin.

For at least the aforementioned reasons, Applicants submit that Cantin in view of NEC does not detract from the patentability of independent claim 23. Claims 24-34 depend on and incorporate the patentable subject matter of independent claim 23. As such, Applicants submit that Cantin in view of NEC does not detract from the patentability of dependent claims 24-34. Accordingly, Applicants respectfully request the withdrawal of the Examiner's rejection of claims 23-34 under 35 U.S.C. §103.

F. Additional Dependent Claim Rejections Under 35 U.S.C. §103

Dependent claims 3-6, 8, 14-17, 19, 25-29 and 34 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC in further view of Hartmut Pohlheim (*"Genetic and Evolutionary Algorithm Toolbox for use with MATLAB").*

Dependent claims 2, 13 and 24 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC in further view of Admitted Prior Art.

Dependent claims 10, 11, 20-22, 31 and 32 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC in view of Hartmut Pohlheim and in further view of Bill Venners ("Eternal Math").

Dependent claim 30 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC in further view of John W. Eaton ("A High Level Interactive Language for Numerical Computations, Edition 3 for Octave Version 2.1x").

Dependent claim 33 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Cantin in view of NEC and in further view of David M. Gay ("Symbolic-Algebraic Computations in a Modeling Language for Mathematical Programming").

None of the cited references, alone or in combination, disclose, teach or suggest each and every feature of independent claims 1, 12 and 23. Claims 2-6, 8, 10 and 11 depend on an incorporate the patentable subject matter of independent claim 1. Claims 13-17, 19, and 20-22 depend on an incorporate the patentable subject matter of independent claim 12. Claims 24, 30-32 and 34 depend on an incorporate the patentable subject matter of independent claim 23. As such, Applicants submit dependent claims 2-6, 8, 10-11, 13-17, 19, 20-22, 24, 30-32 and 34 are patentable and in condition for allowance. Accordingly, Applicants respectfully request the withdrawal of the Examiner's rejection of claims 2-6, 8, 10-11, 13-17, 19, 20-22, 24, 30-32 and 34 under 35 U.S.C. §103.